Developing A Flexible System for Multipathway Environmental Risk Analysis*

David J. LePoire, John J. Arnish, Emmanuel Gnanapragasam, Tim Klett, Bruce M. Biwer, Robert L. Johnson, C. Yu, and S.Y. Chen
Environmental Assessment Division, Argonne National Laboratory

Background and Introduction

Most existing computer codes for environmental pathway modeling were developed to satisfy a specific objective (e.g., perform analyses to demonstrate regulatory compliance). Over time, the codes have been enhanced to assess the impacts to receptors from exposure to multiple pathways (e.g., air, water, and soil). Such capabilities require the ability to model material transfer between different media in addition to the ability to model physical and chemical reactions, dispersion, transport, and uptake. Such previous enhancements were often added without regard to the overall structure of the code, making future expansion difficult. Furthermore, these codes have been written in various computer languages and software environments that are often not compatible with each other. In recent years, largely driven by advances in industrial software development, a new concept for software development based on "modularization" has emerged. This approach entails the development of common "modules" or components that can be shared by and used in different applications that have certain common needs. For instance, an air dispersion model can be written into a common component to be shared by several different applications, each with the need to model air dispersion of some release. When fully developed, the modeling application would become an exercise of selecting, integrating, and applying a consistent combination of appropriate modules for a

*Work supported by the U.S. Department of Energy under Contract No. W-31-109-ENG-38

specific problem. Although modularization promises advantages over the traditional approach, a number of issues do exist. These issues must be fully addressed and resolved before the approach can be accepted as a new paradigm for environmental modeling. This paper discusses these issues in the context of three demonstration projects (LePoire et al. 2001) and provides recommendations and a course of actions for future development.

Traditionally, model connections have been made by the end user, who would align one model's output data with another model's input. Often the model assumptions and conceptualizations were stretched to accomplish the linkage, resulting in greater uncertainty in the results. Also, the connection usually required the user to invest effort in manipulating the data for proper communication (e.g., taking data from the first model's output and manually editing the input for the next level), resulting in inefficient use of resources and introducing another potential source of error. It is generally difficult to connect models because of their disparate assumptions about scale, conceptualization, aggregation, process, reality, and objectives. Systems in other disciplines have been developed by using function libraries and toolboxes to prepare and manipulate data.

Opportunities and Goals

In the environmental field, modeling plays a critical role in connecting current data and knowledge with predictions of future events and environmental states. Environmental problems are quite challenging to solve because of the complex relationships among many contributing factors, both natural and man-made (Constanza et al. 1993).

Moreover, these problems need to be addressed not only by environmental engineers and regulators but also by concerned members of the public and nongovernmental organizations. Their demands on environmental modeling often conflict because predictions need to be accurate yet easily understood, communicated, and explored. The increasing complexity of environmental codes also places a demand on the end user, who must translate the real environmental problem into the limited representation allowed by the model and its options. Information on assumptions and options must be conveyed to the user to ensure that the model is applied and interpreted correctly (Whelan et al. 1997). Open communications about the model, interface, and data components would enable software applications to be more easily developed.

Technology Assessment

If a next-generation risk-modeling environment is to be successful, it must address a range of needs and issues. Identified issues related to the enhanced existing codes include (1) flexibility and maintenance, (2) software dissemination, (3) quality assurance (QA), (4) life-cycle development and maintenance costs, (5) platform reliance (6) transparency, and (7) ease of use.

Various options are available for implementing a more flexible environmental modeling environment. These include (1) continuing with the current status quo approach, (2) adopting a single model-, data-, and user-interface-integrated framework, and (3) using separate tools to integrate models, data, and interfaces (Sydelko et al. 1999). Our proposed solution, adoption of the third approach based on modularity, includes developing strategies and guidelines for separating the software package components into a set of layers and identifying roles for model development and use. The strategies can apply to both the modification of existing codes and practices and the development of new models and components.

Like the traditional approach toward code development, the modularization approach for developing a complete modeling package usually consists of five generally distinct layers: data, model, presentation, application, and network. While the traditional method integrates these layers into a single code, the modularization approach instead aims at building a code system consisting of components that can be used and reused for various purposes.

Three sets of roles are proposed for developing and using the system. First, modelers should develop domain-specific models and document their assumptions. Second, integrators should create an application from the available models and data. The integration environment would be up to the integrator (i.e., there would be no single integration framework, so the system could be done in a web environment [e.g., Active Server Pages or ColdFusion], as a window standalone, or as a hybrid using web services). Third, end users should then specify the data and options through the integrated user interface and communicate the results to the regulators and public.

Technology Demonstrations

Technology options in the various layers (data, model, presentation, application, and network) were explored, and demonstration projects were created to show and evaluate their potential. Traditional software packages are custom integrations of various components (model, data, and user interface). Sometimes the model is somewhat separated from the user interface. Sometimes the data are stored in a flexible format; other times, they are highly formatted and depend on the model. To ensure a chance of integrating software packages, it is important to separate these components. Once the software packages are separated into components, there are many ways to connect them (e.g., in a platform like FRAMES, through a windows development environment like Visual Basic, or in a web-based distributed environment). This flexibility allows developers to share and innovate components for user interfaces, data, models, and network connections, while also allowing incorporation of new technologies.

Three demonstration projects were chosen both to address a current need among radiological analysts and to be potentially useful in later applications. The projects demonstrate the wide variety of integration techniques and ways to use components based on existing software packages, new models, and commercial components.

Low-level landfill analysis:

Using the U.S. Nuclear Regulatory Commission's DUST package (Sullivan 1993) and a modified RESRAD-OFFSITE package (Yu et al. 2001), the models were integrated into a desktop application, with DUST providing a leaching source term to the groundwater and RESRAD providing the multipathway dose assessment from that point forward. The user-interface and model assumptions for both codes were maintained. The integration was made possible because the RESRAD-OFFSITE model has a feature that allows intermediate contaminant fluxes to be output or serve as input for the remainder of the calculation. For this demonstration, the DUST code was used to generate a modeled release flux at the bottom of the landfill. This flux was then used as input to RESRAD-OFFSITE for assessing radiological impacts to a receptor from the groundwater pathways, including the drinking water pathway and pathways associated with the use of contaminated irrigation water. To accomplish this integration, the user interface, model interface, and data components had to be first separated for each model. Then each component was integrated and packaged in a new application. This practice maintained the data integrity, model assumptions, and ease of use.

MARSSIM analysis with RESRAD

MARSSIM is a recent multiagency procedure for finding statistical determinations for radiological cleanup standards. To support MARSSIM activities, RESRAD could be used to generate the dose (or guideline limit) as a function of area. These guidelines could be displayed on a graph and then interpreted on a GIS display of the site with overlain measurements. To demonstrate such capabilities, the RESRAD model was wrapped with a preprocessor and postprocessor for web execution. The pre- and postprocessors allowed simple connections to a customized, simplified, web-based user interface and commercial visualization graphing and geographical information system (GIS) packages.

Nuclide web service

Some research modeling environments use components on a set of widely distributed computers. These distributed computing environments allow standard components to be maintained on a few servers that are optimized for performance and maintained with the current versions. Data and models from various sources can be easily connected, and the data can be communicated across the network. Similarly, radiological risk models often use common methods and sources of information. Rather than duplicate a particular methodology or database for each application, a central repository for such information can be maintained, facilitating periodic updates and reducing QA and maintenance efforts. Nuclide databases are a good example of both a common method and source of information (data). A nuclide web service was developed to demonstrate the feasibility and potential of a distributed system in environmental modeling. In radiological assessment software, the handling of nuclide data is difficult because of the decay chains and different assumptions about secular equilibrium. A simple web service was set up with a limited set of nuclide data to demonstrate their workings. A method that would take input on a radionuclide and deliver decay chain information was developed. The data were recursively extracted from two database tables. One had the nuclide information (e.g., mass, half-life, dose conversion factors, and distribution coefficients for various media). The second detailed the decay relationship, with primary key fields for the parent nuclide and the progeny nuclide and also a field for the yield (or fraction of the decay that followed that decay path).

Future

As new models and packages are developed, some guidelines and standards will help developers design models to be incorporated into larger systems. The separation of data, model, and interface has great potential for new models and applications. Before that happens, existing packages can be separated and wrapped in a manner similar to that demonstrated with RESRAD, RESRAD-OFFSITE, and DUST.

The above projects demonstrated a small part of the potential for component-based environmental modeling with open integration. Components were developed for the user interface, data handling, model wrapping, connection via desktop, web server, and distributed computing environments. An assessment of this system with regard to the issues mentioned is in LePoire et al. (2001).

Many technology uncertainties and risks are addressed by an open system where (1) the modeler and the integrator are separated and (2) the modeling and integration tasks can be done with different tools. Such a system also allows a transition pathway that utilizes existing code with concurrent development of new module code. It also allows the integrator to focus on the user's specific need, whether it is for a detailed analysis without a "big picture" understanding or the ability to navigate around issues while applying regulatory requirements to analyze a specific site.

However, there are some drawbacks. Sometimes the technology can be under such rapid development that an integration system might depend on a commercial tool that is supported for only a short amount of time. It is hoped that the components could be developed to be flexible enough so they could be easily modified to function with the latest systems. In addition, the environmental modeling situation is quite complex because of the range of stakeholders involved in environmental problems. These include government agencies, regulators, end users, model developers, integration developers, and public citizens and organizations.

Recommendations

Component-based environmental modeling offers many advantages as long as the hazards in developing the system are dealt with. An open system of components and integration techniques offers the hope of addressing issues in an open and shared environment to leverage existing codes in multiple integrations. An open system allows the sharing of models, data, and interface components for many integration techniques.

On the basis of the above discussion, the following four recommendations are made: (1) extend interagency discussions (Whelan et al. 2001) to address these flexible future model needs and issues, (2) maximize the use of technologies developed by the software industry, (3) maintain the integrity of legacy codes, and (4) minimize dependence on a particular system.

References

Constanza R, Wainger L, Folke C, Maler KG, Modeling complex ecological economic systems. BioScience 43; September 1993.

LePoire DJ, Arnish JJ, Gnanapragasam E, Klett T, Johnson RL, Chen SY, Biwer BM, Yu C. OpenLink: A flexible integration system for environmental risk analysis and management. ANL/EAD/TM-114; Argonne, IL: Argonne National Laboratory; October 2001.

Sullivan TM. Disposal unit source term (DUST) data input guide. NUREG/CR-6041; BNL-NUREG-52375; May 1993.

Sydelko PJ, Majerus KA, Dolph JA, Taxon TN. A dynamic object-oriented architecture approach to ecosystem modeling and simulation. Proceedings of the 1999 American Society of Photogammetry and Remote Sensing (ASPRS) annual conference, Portland, OR; May 19-21, 1999: 410-421.

Whelan G, Pelton MA, Castleton KJ, Strenge DJ, Buck JW, Gelston GM, Hoopes BL, Kickert RN. Concepts of a framework for risk analysis in multimedia environmental systems; Richland, WA: Pacific Northwest National Laboratory; October 1997.

Whelan G, Nicholson T (editors). Proceedings of the environmental software systems compatibility and linkage workshop. Richland, WA: Pacific Northwest National Laboratory; in press.

Yu C, Zielen AJ, Cheng JJ, LePoire DJ, Gnanapragasam E, Kamboj S, Arnish JJ, Wallo III A, Williams WA, Peterson H. User's manual for RESRAD version 6; Argonne IL: Argonne National Laboratory; ANL/EAD-4, July 2001.